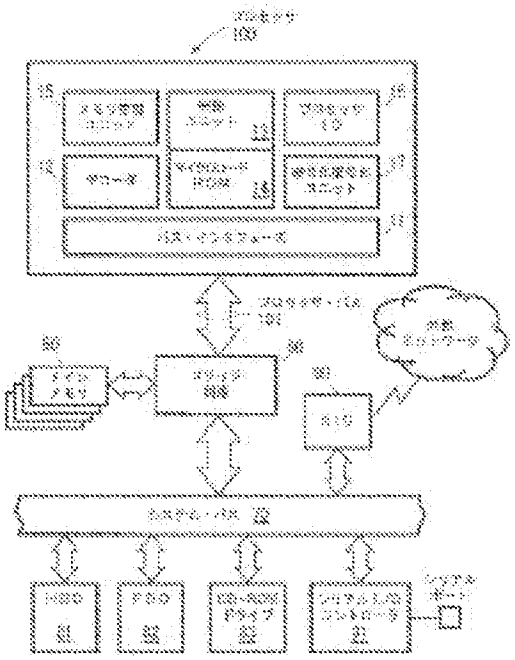
PROCESSOR WITH UNAUTHORIZED EXECUTION PREVENTING FUNCTION OF PROGRAM, INSTRUCTION TO BE EXECUTED BY PROCESSOR AND UNAUTHORIZED EXECUTION PREVENTING METHOD OF THE PROGRAM

Publication number: JP11345117 (A)
Publication date: 1999-12-14
Inventor(s): OKADA TAKAYUKI +
Applicant(s): IBM +
Classification:
- international: ***G06F21/00; G06F21/22;*** **G06F21/00; G06F21/22;** (IPC1-7): G06F9/06
- European: G06F21/00N7P5H
Application number: JP19980136724 19980519
Priority number(s): JP19980136724 19980519

Also published as:
JP3713141 (B2)
US6704872 (B1)

Abstract of **JP 11345117 (A)**

PROBLEM TO BE SOLVED: To provide an improved techniques for preventing unauthorized use/ execution of a software program provided to a computer system. SOLUTION: A processor 100 incorporates a memory managing unit 15 for managing the storage operation of code/data on a main memory and is provided with an execution permitting instruction for permitting the execution of an application on the main memory as an instruction set. The procedure for the execution permitting instruction is defined by a microprogram and provided with a procedure for certifying the right of using a program. Since the certifying operation is executed inside the processor, it is nearly impossible to monitor/ detect operation from the outside.; Besides by using a processor ID 16 characteristically provided in each processor chip as identification information to be used for certifying processing, the right of using the program can be limited to a single processor.



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平11－345117

(43)公開日　平成11年(1999)12月14日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　9/06	５５０	Ｃ０６Ｆ　9/06	５５０Ｈ

審査請求　未請求　請求項の数18　ＯＬ　（全 21 頁）

(21)出願番号　特願平10－136724

(22)出願日　平成10年(1998) 5月19日

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレイション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国10504、ニューヨーク州　アーモンク　（番地なし）

(72)発明者　岡田　高幸
滋賀県野洲郡野洲町大字市三宅800番地　日本アイ・ビー・エム株式会社　野洲事業所内

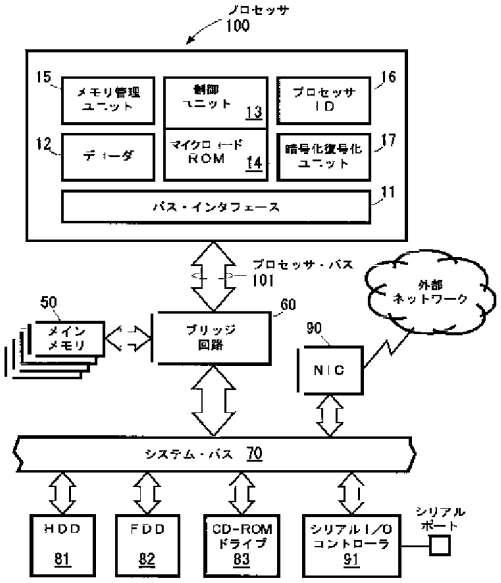
(74)代理人　弁理士　坂口　博　（外１名）

(54)【発明の名称】　プログラムの不正実行防止機能付きプロセッサ、プロセッサが実行するインストラクション、及びプログラムの不正実行防止方法

(57)【要約】　（修正有）

【課題】　コンピュータ・システムに提供されるソフトウェア・プログラムの不正な使用・実行を防止する優れた技術を提供する。

【解決手段】　プロセッサは、メイン・メモリ上のコード／データの格納動作を管理するためのメモリ管理ユニットを内蔵し、かつインストラクション・セットとして、メイン・メモリ上のアプリケーションの実行を許可する実行許可インストラクションを含んでいる。実行許可インストラクションの手続はマイクロプログラムによって定義され、プログラムの使用権を認証するための手続を含んでいる。認証動作は、プロセッサ内部で実行されるので、外部から監視・検出することは殆ど不可能である。また、認証処理に用いる識別情報として、プロセッサ・チップが固有に持つプロセッサＩＤを使用することによって、プログラムの使用権を単一のプロセッサに限定することができる。



【特許請求の範囲】
【請求項１】メイン・メモリを管理する機能を持つメモリ管理ユニットを内蔵するとともに、自身を動作させるためのインストラクション・セットを有するタイプの、プログラムの不正実行防止機能付きプロセッサであって、
前記インストラクション・セット中には前記メイン・メモリ上のプログラム・コードの実行を許可する実行許可インストランションが含まれ、
前記実行許可インストラクションは、実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作の他に、所定の認証動作を含む、ことを特徴とするプログラムの不正実行防止機能付きプロセッサ。
【請求項２】前記所定の認証動作は、前記プロセッサ内部に格納された識別情報と前記プロセッサの外部から供給された識別情報との比較動作を含むことを特徴とする請求項１に記載のプログラムの不正実行防止機能付きプロセッサ。
【請求項３】前記識別情報は、前記プロセッサに固有のプロセッサＩＤとソフトウェア・プログラムに固有のソフトウェアＩＤとで構成されることを特徴とする請求項２に記載のプログラムの不正実行防止機能付きプロセッサ。
【請求項４】前記識別情報は、前記プロセッサが内部的に発生した乱数とソフトウェア・プログラムに固有のソフトウェアＩＤとで構成されることを特徴とする請求項２に記載のプログラムの不正実行防止機能付きプロセッサ。
【請求項５】前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を禁止することを特徴とする請求項１に記載のプログラムの不正実行防止機能付きプロセッサ。
【請求項６】メイン・メモリを管理するためのメモリ管理ユニットを内蔵したタイプのプロセッサが実行するインストラクションであって、
前記インストラクションは、アプリケーションの実行を許可するためのインストラクションであり、
所定の認証動作と、
実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作と、を特徴とするプロセッサが実行するインストラクション。
【請求項７】前記所定の認証動作は、前記プロセッサ内部に格納された識別情報と前記プロセッサの外部から供給された識別情報との比較動作を含むことを特徴とする請求項６に記載のプロセッサが実行するインストラクション。
【請求項８】前記識別情報は、前記プロセッサに固有のプロセッサＩＤとソフトウェア・プログラムに固有のソフトウェアＩＤとで構成されることを特徴とする請求項７に記載のプロセッサが実行するインストラクション。
【請求項９】前記識別情報は、前記プロセッサが内部的に発生した乱数とソフトウェア・プログラムに固有のソフトウェアＩＤとで構成されることを特徴とする請求項７に記載のプロセッサが実行するインストラクション。
【請求項１０】前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を禁止することを特徴とする請求項６に記載のプロセッサが実行するインストラクション。
【請求項１１】メイン・メモリ上にプログラム・コードやデータを読み出し／書き込み動作しながら所定の処理を実行するタイプの、プログラムの不正実行防止機能付きプロセッサであって、
前記プロセッサの外部とインストラクションやデータの授受を行うバス・インターフェースと、
受け取ったインストラクションの解読を行うデコーダと、
１以上のインストラクションについての手順を格納した内蔵メモリと、
１以上のインストラクションについての手順を実装した論理回路と、
前記内蔵メモリに格納された手順や前記論理回路に実装された手順に従ってインストラクションを実行する制御ユニットと、
前記プロセッサに固有のプロセッサＩＤと、
前記メイン・メモリを管理する機能を持つメモリ管理ユニットとを含み、
前記内蔵メモリ又は前記論理回路のうちの１つは前記メイン・メモリ上のプログラム・コードの実行を許可する実行許可インストラクションの手順を格納しており、
且つ、前記実行許可インストラクションの手順は、実行すべきプログラムの識別情報とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作の他に、所定の認証動作を含む、ことを特徴とするプログラムの不正実行防止機能付きプロセッサ。
【請求項１２】さらに、前記プロセッサＩＤを暗号化して前記バス・インターフェース経由で前記プロセッサの外部に出力する暗号化手段と、暗号化された識別情報を前記バス・インターフェース経由で前記プロセッサの外部から受け取るとともに復号化してプロセッサＩＤとソフトウェアＩＤを取り出す復号化手段とを含み、
前記所定の認証動作は各ＩＤ同士の比較動作を含むこと

を特徴とする請求項１１に記載のプログラムの不正実行防止機能付きプロセッサ。
【請求項１３】前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を禁止することを特徴とする請求項１１に記載のプログラムの不正実行防止機能付きプロセッサ。
【請求項１４】メイン・メモリ上にプログラム・コードやデータを読み出し／書き込み動作しながら所定の処理を実行するタイプの、プログラムの不正実行防止機能付きプロセッサであって、
前記プロセッサの外部とインストラクションやデータの授受を行うバス・インターフェースと、
受け取ったインストラクションの解読を行うデコーダと、
１以上のインストラクションについての手順を格納した内蔵メモリと、
１以上のインストラクションについての手順を実装した論理回路と、
前記内蔵メモリに格納された手順や前記論理回路に実装された手順に従ってインストラクションを実行する制御ユニットと、
乱数発生ユニットと、
前記メイン・メモリを管理する機能を持つメモリ管理ユニットとを含み、
前記内蔵メモリは前記メイン・メモリ上のプログラム・コードの実行を許可する実行許可インストラクションの手順を格納しており、
且つ、前記実行許可インストラクションの手順は、実行すべきプログラムの識別情報とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作の他に、所定の認証動作を含む、ことを特徴とするプログラムの不正実行防止機能付きプロセッサ。
【請求項１５】さらに、前記乱数発生ユニットが発生する乱数を前記バス・インターフェース経由で前記プロセッサの外部に出力する乱数出力手段と、暗号化された識別情報を前記バス・インターフェース経由で前記プロセッサの外部から受け取るとともに復号化して乱数とソフトウェアＩＤを取り出す復号化手段とを含み、
前記所定の認証動作は乱数及びソフトウェアＩＤの比較動作を含むことを特徴とする請求項１４に記載のプログラムの不正実行防止機能付きプロセッサ。
【請求項１６】前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を禁止することを特徴とする請求項１４に記載のプログラムの不正実行防止機能付きプロセッサ。
【請求項１７】ソフトウェア供給者がソフトウェア・プログラムの使用権をただ１つのプロセッサにのみ許可するための、プログラムの不正実行防止方法であって、
（ａ）ソフトウェアＩＤが付されたソフトウェア・プログラムを受容するステップと、（ｂ）前記プロセッサに固有のプロセッサＩＤを暗号化して、前記プロセッサの外部に出力するステップと、（ｃ）ソフトウェア供給者において、暗号化されたプロセッサＩＤを一旦復号化して、ソフトウェア供給者自身が管理するソフトウェアＩＤを前記プロセッサＩＤとともに暗号化するステップと、（ｄ）ステップ（ｃ）によって暗号化した識別情報を前記プロセッサに供給するステップと、（ｅ）前記プロセッサにおいて、識別情報を復号化してプロセッサＩＤとソフトウェアＩＤを取り出すステップと、（ｆ）前記プロセッサが保持するプロセッサＩＤ及び前記ソフトウェア・プログラムに付されたソフトウェアＩＤを、ステップ（ｅ）で得られたプロセッサＩＤとソフトウェアＩＤの各々と照合するステップと、（ｇ）前記照合処理が成功裡に終わったときのみ、前記プロセッサが前記ソフトウェア・プログラムを実行することを許可するステップと、を具備することを特徴とするプログラムの不正実行防止方法。
【請求項１８】ソフトウェア供給者がソフトウェア・プログラムの使用権をただ１つのプロセッサにのみ許可するための、プログラムの不正実行防止方法であって、
（ａ）ソフトウェアＩＤが付されたソフトウェア・プログラムを受容するステップと、（ｂ）前記プロセッサが乱数を発生して前記プロセッサの外部に出力するステップと、（ｃ）ソフトウェア供給者において、ソフトウェア供給者自身が管理するソフトウェアＩＤを前記プロセッサが出力した乱数とともに暗号化するステップと、
（ｄ）ステップ（ｃ）によって暗号化した識別情報を前記プロセッサに供給するステップと、（ｅ）前記プロセッサにおいて、識別情報を復号化して乱数とソフトウェアＩＤを取り出すステップと、（ｆ）前記プロセッサが保持する乱数及び前記ソフトウェア・プログラムに付されたソフトウェアＩＤを、ステップ（ｅ）で得られた乱数とソフトウェアＩＤの各々と照合するステップと、
（ｇ）前記照合処理が成功裡に終わったときのみ、前記プロセッサが前記ソフトウェア・プログラムを実行することを許可するステップと、を具備することを特徴とするプログラムの不正実行防止方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、コンピュータ・システムに提供されるソフトウェア・プログラムの不正な使用・実行を防止する技術に係り、特に、プログラムの

不正実行防止機能を備えたプロセッサ、プログラムの不正実行を防止するためにプロセッサが実行するインストラクション、及びプログラムの不正実行防止方法に関する。更に詳しくは、本発明は、外部から監視・変更することができないプログラムの不正実行防止機能付きプロセッサ、プロセッサが実行するインストラクション、及びプログラムの不正実行防止方法に関する。

【０００２】

【従来の技術】昨今の技術革新に伴い、コンピュータ・システムは、大学・研究機関の他、企業や一般家庭にも広範に浸透してきている。コンピュータ・システムには、ホスト／メインフレームやオフィス・コンピュータ以外に、比較的安価で一般ユーザも購入可能なワークステーション、パーソナル・コンピュータ（ＰＣ）などが挙げられる。特に、ＰＣの急速な普及には目覚ましいものがある。

【０００３】このような技術動向下では、コンピュータ・システム上で稼働する各種ソフトウェアも開発・市販されており、ソフトウェア産業もハードウェア産業に匹敵しあるいは凌駕する隆盛を遂げている。ここで言うソフトウェアには、コンピュータ・システム全体の動作を制御するための「ＯＳ（オペレーティング・システム）」や、システム上で各ユーザの目的・業務に適った機能オペレーションを実現するための「アプリケーション」が含まれる。アプリケーション・プログラムは、ワープロ、スプレッドシート、データベース、通信など各種機能・用途のものが揃っている。

【０００４】ユーザは、自身のコンピュータ・システム上に導入したいソフトウェアを、例えばフロッピー・ディスクやＣＤ－ＲＯＭなどのリムーバブル・メディアの形態で購入し、これらメディアを適切なドライブ・ユニットに装填して用いるのが一般的である。また、最近では、インターネットなどの外部ネットワークを経由して、所望のソフトウェアを自身のローカル・ディスクにファイル転送（ダウンロード）する、という形態によってもソフトウェア・プログラムを導入することが可能となっている。

【０００５】但し、ソフトウェア・プログラムを格納したメディアの購入行為は、ソフトウェア・プログラムの限定的な使用権を得たことに過ぎず、購入者はソフトウェア・プログラムの著作権や複製権までも手に入れた訳ではない。ソフトウェア・プログラムの無制限な若しくは無断な複製は、著作権法上、厳に禁止されている行為である（周知）。（なお、ネットワーク経由でダウンロードしたソフトウェア・プログラムについても、その使用は限定的に解釈するのが妥当であろう。）

【０００６】また、ソフトウェア・プログラムを提供するソフトウェア・ベンダーの立場から言えば、有料でプログラム格納メディアを配布することによって収益をあげているのであり、無制限で無断な複製行為が横行していては、本来享受すべき利潤を担保することができない。ソフトウェア・プログラムの不正な使用行為は、ソフトウェア産業に従事する者の開発意欲をも減退させるものであり、ひいては当該産業界の活動自体を沈静化させかねない。このため、ソフトウェア・プログラムの不正使用や無断な複製を禁止する（あるいは未然に防止する）というプロテクションに関する技術が、従来より開発されている。

【０００７】プロテクションの一例は、不正使用を防止したいソフトウェア・プログラムを暗号化してしまうことである。すなわち、ソフトウェア・ベンダーは、暗号化した状態のプログラム格納メディアを有償若しくは無償で配布するとともに、正当なユーザに対してのみ、暗号を解くための鍵を与えることで、メディアに格納されたプログラムの使用を好適に限定することができる訳である。

【０００８】ここで、いかなる媒介によってユーザに鍵を与えるかが問題になる。例えばソフトウェア・ベンダーは、封書などの郵便物によってユーザに鍵を与えることも可能であるが、この場合は鍵の不正使用はいとも簡単である。

【０００９】さらに技術的に高度な方式として、いわゆる「セキュリティ・デバイス」を用いることが挙げられる。この場合、ソフトウェア・ベンダーは、セキュリティ・デバイスを添付してプログラム格納メディアを流通・配布する。セキュリティ・デバイスは、所定の認証手続に必要な識別情報などを含んでおり、例えばユーザのＰＣのシリアル・ポート又はパラレル・ポートに接続して用いられる。他方、メディアに格納されたプログラムの中には、プログラム自体の機能・用途に特化したオペレーションを行うための本ルーチンの他に、幾つかのチェック・ポイント（すなわち「認証ルーチン」）を含んでいる。

【００１０】認証ルーチンの一例は、各チェック・ポイントにおいて、プログラムを実行中のＣＰＵ（Central Processing Unit）がセキュリティ・デバイスにアクセスして識別情報を読み出し、これがプログラムの持つ識別情報と一致するかどうかを照合して、一致しているときのみＣＰＵがプログラムを続行することを許可する、というものである。

【００１１】また、認証ルーチンの他の例は次の通りである。すなわち、プログラム実行中のＣＰＵがある規則に従ったコードをセキュリティ・デバイスに書き込む。セキュリティ・デバイスでは、受け取ったコードを用いてスクランブル又は暗号化した識別情報を用意しておく。さらに、ＣＰＵはスクランブル又は暗号化された識別情報を読み出し、これをプログラムによってデスクランブル又は復号化してプログラムが所有する識別情報と照合する。照合が成功裡に終わったときのみ、ＣＰＵによるプログラムの続行が許可される。ＣＰＵとセキュリ

ティ・デバイスの間は、ＣＰＵチップ外部のバスを介して接続されているので、スクランブルや暗号化を行うことによって、バス・スヌープによる認証ルーチンの解読を防止することができる。

【００１２】これらセキュリティ・デバイスを用いた２つの例は、プログラム実行中のＣＰＵに対する命令（インストラクション）によって認証処理が行われる、という点では共通している。

【００１３】セキュリティ・デバイスを用いることにより、セキュリティ・レベルをかなりの程度向上させることができよう。また、ソフトウェア・プログラムの使用権をセキュリティ・デバイスを装着したただ１つのコンピュータ・システムのみに限定することができよう。しかしながら、この方式とて万全とは言い難い。例えば、ロジック・アナライザ（周知）を用いれば、バス・トランザクションを容易にモニタすることができるので、認証ルーチンを解析して、識別情報や暗号の鍵の正体を見破ることもできよう。

【００１４】さらにセキュリティ・レベルを向上させるために、セキュリティ・デバイスへの入力値を逐次変更する、という対策も採られている。しかしながら、ＩＣＥ（InCircuitEmulator）/*/などの装置を用いればソフトウェアを比較的容易に逆アセンブルすることができ、そうなればセキュリティ・デバイスへのアクセス・ルーチン（すなわち認証ルーチン）を無条件ジャンプするようにプログラムを改ざんすることも可能である。

【００１５】本発明者の考察によれば、ソフトウェア・プログラムのプロテクションを強化するためには、まず第一に、セキュリティ動作がプログラム実行に必須であるようにデザインすることが好ましい。この意味において、メディアに格納されたプログラム自身が認証ルーチンを含んでいるという上記従来例は適合している。さらに、この認証ルーチンは、外部からのアクセス・変更が不可能な形態でなければならない。セキュリティ・デバイスを用いた上記従来例の場合、認証ルーチンはシステム・オペレーションの形態で実装され、外観には現れないが、技術的には判読が可能である（前述）。要言すれば、外部アクセス可能な従来のプロテクション方式では、プログラムの保護は万全とは言い難いのである。

【００１６】《注釈》ＩＣＥは、通常はプログラムやハードウェア開発を支援するために使用される装置であり、ＣＰＵの代わりにＣＰＵソケットに差すことによって、ＣＰＵと全く同じ動作をするようになっている。ＩＣＥは、ＣＰＵと違って、バスのアクセスやレジスタの値をインストラクション毎に読み出すことが可能である。このようなＩＣＥのインストラクション・トレース機能を用いれば、外部デバイスのアドレスをトリガにしてソフトウェア・プログラムの中から認証ルーチンを探し出すことができる。さらに、プログラム・アドレスに違法なパッチを当てれば、セキュリティ・デバイスにアクセスせずにプログラム実行を継続するように改ざんすることができる。

【００１７】

【発明が解決しようとする課題】本発明の目的は、コンピュータ・システムに提供されるソフトウェア・プログラムの不正な使用・実行を防止する優れた技術を提供することにある。

【００１８】本発明の更なる目的は、プログラムの不正実行防止機能を備えた優れたプロセッサ、プログラムの不正実行を防止するためにプロセッサが実行するインストラクション、及びプログラムの不正実行防止方法を提供することにある。

【００１９】本発明の更なる目的は、外部から監視・変更することができないプログラムの不正実行防止機能付き優れたプロセッサ、プロセッサが実行するインストラクション、及びプログラムの不正実行防止方法を提供することにある。

【００２０】

【課題を解決するための手段】本発明は、上記課題を参酌してなされたものであり、その第１の側面は、メイン・メモリを管理する機能を持つメモリ管理ユニットを内蔵するとともに、自身を動作させるためのインストラクション・セットを有するタイプの、プログラムの不正実行防止機能付きプロセッサであって、前記インストラクション・セット中には前記メイン・メモリ上のプログラム・コードの実行を許可する実行許可インストランションが含まれ、前記実行許可インストラクションは、実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作の他に、所定の認証動作を含む、ことを特徴とするプログラムの不正実行防止機能付きプロセッサである。

【００２１】ここで、前記所定の認証動作は、前記プロセッサ内部に格納された識別情報と前記プロセッサの外部から供給された識別情報との比較動作を含んでいてもよい。

【００２２】認証に用いられる識別情報は、前記プロセッサに固有のプロセッサＩＤとソフトウェア・プログラムに固有のソフトウェアＩＤとで構成されていてもよい。

【００２３】あるいは、識別情報は、前記プロセッサが内部的に発生した乱数とソフトウェア・プログラムに固有のソフトウェアＩＤとで構成されていてもよい。

【００２４】また、前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を禁止するようにしてもよい。

【００２５】また、本発明の第２の側面は、メイン・メ

モリを管理するためのメモリ管理ユニットを内蔵したタイプのプロセッサが実行するインストラクションであって、前記インストラクションは、アプリケーションの実行を許可するためのインストラクションであり、所定の認証動作と、実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作と、を特徴とするプロセッサが実行するインストラクションである。

【００２６】ここで、前記所定の認証動作は、前記プロセッサ内部に格納された識別情報と前記プロセッサの外部から供給された識別情報との比較動作を含んでいてもよい。

【００２７】認証に用いられる識別情報は、前記プロセッサに固有のプロセッサＩＤとソフトウェア・プログラムに固有のソフトウェアＩＤとで構成されていてもよい。

【００２８】あるいは、識別情報は、前記プロセッサが内部的に発生した乱数とソフトウェア・プログラムに固有のソフトウェアＩＤとで構成されていてもよい。

【００２９】また、前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を禁止するようにしてもよい。

【００３０】また、本発明の第３の側面は、メイン・メモリ上にプログラム・コードやデータを読み出し／書き込み動作しながら所定の処理を実行するタイプの、プログラムの不正実行防止機能付きプロセッサであって、前記プロセッサの外部とインストラクションやデータの授受を行うバス・インターフェースと、受け取ったインストラクションの解読を行うデコーダと、１以上のインストラクションについての手順を格納した内蔵メモリと、１以上のインストラクションについての手順を実装した論理回路と、前記内蔵メモリに格納された手順や前記論理回路に実装された手順に従ってインストラクションを実行する制御ユニットと、前記プロセッサに固有のプロセッサＩＤと、前記メイン・メモリを管理する機能を持つメモリ管理ユニットとを含み、前記内蔵メモリ又は前記論理回路のうちの１つは前記メイン・メモリ上のプログラム・コードの実行を許可する実行許可インストラクションの手順を格納しており、且つ、前記実行許可インストラクションの手順は、実行すべきプログラムの識別情報とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作の他に、所定の認証動作を含む、ことを特徴とするプログラムの不正実行防止機能付きプロセッサ。

【００３１】第３の側面に係る不正実行防止機能付きプロセッサは、さらに、前記プロセッサＩＤを暗号化して前記バス・インターフェース経由で外部に出力する暗号化手段と、暗号化された識別情報を前記バス・インターフェース経由で外部から受け取るとともに復号化してプロセッサＩＤとソフトウェアＩＤを取り出す復号化手段とを含み、前記所定の認証動作は各ＩＤ同士の比較動作を含んでいてもよい。

【００３２】また、前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を禁止するようにしてもよい。

【００３３】また、本発明の第４の側面は、メイン・メモリ上にプログラム・コードやデータを読み出し／書き込み動作しながら所定の処理を実行するタイプの、プログラムの不正実行防止機能付きプロセッサであって、前記プロセッサの外部とインストラクションやデータの授受を行うバス・インターフェースと、受け取ったインストラクションの解読を行うデコーダと、１以上のインストラクションについての手順を格納した内蔵メモリと、１以上のインストラクションについての手順を実装した論理回路と、前記内蔵メモリに格納された手順や前記論理回路に実装された手順に従ってインストラクションを実行する制御ユニットと、乱数発生ユニットと、前記メイン・メモリを管理する機能を持つメモリ管理ユニットとを含み、前記内蔵メモリ又は前記論理回路のうちの１つは前記メイン・メモリ上のプログラム・コードの実行を許可する実行許可インストラクションの手順を格納しており、且つ、前記実行許可インストラクションの手順は、実行すべきプログラムの識別情報とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作の他に、所定の認証動作を含む、ことを特徴とするプログラムの不正実行防止機能付きプロセッサである。

【００３４】第４の側面に係る不正実行防止機能付きプロセッサは、さらに、前記乱数発生ユニットが発生する乱数を前記バス・インターフェース経由で外部に出力する乱数出力手段と、暗号化された識別情報を前記バス・インターフェース経由で外部から受け取るとともに復号化して乱数とソフトウェアＩＤを取り出す復号化手段とを含み、前記所定の認証動作は乱数及びソフトウェアＩＤの比較動作を含んでいてもよい。

【００３５】また、前記所定の認証動作が成功裡に終わったときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を許可するが、前記所定の認証動作が失敗したときには実行すべきプログラムの識別番号とアドレス情報・属性情報を前記メモリ管理ユニットに与える動作を禁止するようにしてもよい。

【００３６】また、本発明の第５の側面は、ソフトウェ

ア供給者がソフトウェア・プログラムの使用権をただ１つのプロセッサにのみ許可するための、プログラムの不正実行防止方法であって、（ａ）ソフトウェアＩＤが付されたソフトウェア・プログラムを受容するステップと、（ｂ）前記プロセッサに固有のプロセッサＩＤを暗号化して、前記プロセッサの外部に出力するステップと、（ｃ）ソフトウェア供給者において、暗号化されたプロセッサＩＤを一旦復号化して、ソフトウェア供給者自身が管理するソフトウェアＩＤを前記プロセッサＩＤとともに暗号化するステップと、（ｄ）ステップ（ｃ）によって暗号化した識別情報を前記プロセッサに供給するステップと、（ｅ）前記プロセッサにおいて、識別情報を復号化してプロセッサＩＤとソフトウェアＩＤを取り出すステップと、（ｆ）前記プロセッサが保持するプロセッサＩＤ及び前記ソフトウェア・プログラムに付されたソフトウェアＩＤを、ステップ（ｅ）で得られたプロセッサＩＤとソフトウェアＩＤの各々と照合するステップと、（ｇ）前記照合処理が成功裡に終わったときのみ、前記プロセッサが前記ソフトウェア・プログラムを実行することを許可するステップと、を具備することを特徴とするプログラムの不正実行防止方法である。

【００３７】また、本発明の第６の側面は、ソフトウェア供給者がソフトウェア・プログラムの使用権をただ１つのプロセッサにのみ許可するための、プログラムの不正実行防止方法であって、（ａ）ソフトウェアＩＤが付されたソフトウェア・プログラムを受容するステップと、（ｂ）前記プロセッサが乱数を発生して前記プロセッサの外部に出力するステップと、（ｃ）ソフトウェア供給者において、ソフトウェア供給者自身が管理するソフトウェアＩＤを前記プロセッサが出力した乱数とともに暗号化するステップと、（ｄ）ステップ（ｃ）によって暗号化した識別情報を前記プロセッサに供給するステップと、（ｅ）前記プロセッサにおいて、識別情報を復号化して乱数とソフトウェアＩＤを取り出すステップと、（ｆ）前記プロセッサが保持する乱数及び前記ソフトウェア・プログラムに付されたソフトウェアＩＤを、ステップ（ｅ）で得られた乱数とソフトウェアＩＤの各々と照合するステップと、（ｇ）前記照合処理が成功裡に終わったときのみ、前記プロセッサが前記ソフトウェア・プログラムを実行することを許可するステップと、を具備することを特徴とするプログラムの不正実行防止方法である。

【００３８】

【作用】本発明に係るプロセッサは、メイン・メモリ上のコード／データの格納動作を管理するためのメモリ管理ユニットを内蔵するタイプであり、且つ、インストラクション・セットの１つとして、メイン・メモリ上のプログラム・コード（アプリケーション）の実行を許可する実行許可インストラクションを含んでいる。

【００３９】この種の実行許可インストラクションは、一般には、実行すべきプログラムの識別番号と、プログラムの格納アドレス、プログラムの属性情報をメモリ管理ユニットにセットする、という手続（例えば"LOAD MMU …"というコードで記述される）を含んでいる。インストラクションの手続はマイクロプログラムによって定義される。

【００４０】本発明では、この実行許可インストラクションは、メモリ管理ユニットに所定のデータをセットする手続の他に、プログラムの使用権を認証するための認証手続を含んでいる。この認証動作は、例えば、プロセッサ内部で保持された識別情報と、プロセッサの外部（例えばソフトウェア・プログラムの供給者）から供給された識別情報との比較によって行われる。そして、この認証処理が成功裡に終わったときのみ、ソフトウェア・プログラムの実行が許可される。

【００４１】認証手続に用いる識別情報は、例えば、プロセッサに固有（例えば製造時に与えられる）プロセッサＩＤと、ソフトウェア・プログラムに固有に与えられるソフトウェアＩＤとで構成される。あるいは、識別情報は、プロセッサが時々刻々発生する乱数と、ソフトウェア・プログラムに固有に与えられるソフトウェアＩＤとで構成される。

【００４２】このような認証動作は、プロセッサ内部で実行されるので、外部から監視・検出することは殆ど不可能である。また、認証手続は、プログラムの実行許可を意味するインストラクションを記述したマイクロプログラムの中に含まれているので、そもそも認証手続の存在自体が外部から隠された格好となる。したがって、本発明によればプログラムの不正実行の防止を大幅に強化することができる。

【００４３】また、認証処理に用いる識別情報として、プロセッサ・チップが固有に持つプロセッサＩＤを使用することによって、ある特定のソフトウェア・プログラムの使用権を単一のプロセッサに限定することができる。この意味においても、プログラムの不正実行は強固に禁止されていると言えよう。

【００４４】本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

【００４５】

【発明の実施の形態】以下、図面を参照しながら本発明の実施例を詳解する。

【００４６】第１の実施の形態

図１には、本発明の第１の実施形態に係るコンピュータ・システム２００及びプロセッサ１００のハードウェア構成を模式的に示している。コンピュータ・システム２００は、システム２００の中核的な制御を行うプロセッサ１００の外に、メイン・メモリ５０や各種周辺装置を含んでいる。以下、各部について説明する。

【００４７】プロセッサ１００の外部ピンに直結したプロセッサ・バス１０１は、ブリッジ回路６０経由でシステム・バス７０に接続されている。ブリッジ回路６０は、プロセッサ・バス１０１とシステム・バス７０間の動作速度の相違を吸収するためのデータ・バッファや、メイン・メモリ５０へのアクセス動作を制御するためのメモリ・コントローラを含んでいる。

【００４８】メイン・メモリ５０は、プロセッサ１００が実行するプログラム・コードや処理データを一時格納するための読み書き可能メモリであり、通常は、１以上のＤＲＡＭ（ダイナミックＲＡＭ）チップで構成される。メイン・メモリ５０上で展開されるメモリ空間は、メモリ管理ユニット１５（後述）によって管理される。

【００４９】システム・バス７０には、ＰＣＩ（Peripheral Component Interconnect）バスやＩＳＡ（Industry Standard Architecture）バスが該当し、各種周辺装置が相互接続されている。周辺装置には、例えばキーボードやディスプレイなどのコンソール類（図示しない）、ハード・ディスク・ドライブ（ＨＤＤ）８１、フロッピー・ディスク・ドライブ（ＦＤＤ）８２、ＣＤ－ＲＯＭドライブ８３などの外部記憶装置類、外部ネットワークと接続するためのネットワーク・インターフェース・カード（ＮＩＣ）９０、シリアル・ポート経由でシリアル・データの入出力を行うためのシリアルＩ／Ｏコントローラ９１などが含まれる。なお、バス７０上の動作は、ロジック・アナライザなどを用いて解析することができる。

【００５０】ソフトウェア供給業者は、通常、フロッピー・ディスクやＣＤ－ＲＯＭなどの記憶媒体の形態でソフトウェア・プログラムを一般ユーザに配布する。各ユーザは、これら記憶媒体を所定のドライブ・ユニットに装填し、ＨＤＤ８１にコピーすることによって、ソフトウェア・プログラムをシステム２００にインストールすなわち利用可能な状態に置くようになっている。また、最近では、インターネットなどの外部ネットワーク経由でソフトウェア・プログラムが配布されることも多くなってきている（周知）。

【００５１】プロセッサ１００は、当業界においてＣＰＵ（Central Processing Unit）若しくはＭＰＵ（Micro Processing Unit）と呼ばれるものである。プロセッサ１００は、通常はマザー・ボード（図示しない）上に搭載される回路チップであり、各入出力装置（図示しない）を制御してデータの授受を行ったり、データを演算処理し、処理結果をメイン・メモリ５０上に記憶したりするなどのコンピュータ・システム２００の中枢機能を果たしている。

【００５２】図１に示すように、本実施形態に係るプロセッサ１００は、バス・インターフェース１１と、デコーダ１２と、制御ユニット１３と、マイクロコードＲＯＭ１４と、メモリ管理ユニット１５と、プロセッサＩＤ１６と、暗号化復号化ユニット１７とを含んでいる。

【００５３】バス・インターフェース１１は、プロセッサ１００外部のコンポーネントとの間でインストラクションやデータを授受するためのユニットである。バス・インターフェース１１は、プロセッサ・バス１７と直結している。なお、プロセッサ・バス１７は、ブリッジ回路６０を経由してメイン・メモリ５０やシステム・バス７０と相互接続されている。

【００５４】デコーダ１２は、バス・インターフェース１１経由で受け取ったインストラクションを解釈して、制御ユニット１３が理解できる形式に変換するためのユニットである。

【００５５】制御ユニット１３は、インストラクションを実行する手順を整えるユニットである。実行するインストラクションは、ワイヤード・ロジック（配線論理）で実行するタイプと、マイクロ・プログラムで実行するタイプの２つに大別される。前者の場合は、インストラクションの実行制御に必要な論理を直線的に実装した順序回路が用いられる。また、後者の場合には、インストラクションはマイクロコード（又はピココード）と呼ばれるさらに細かいコードの集合（すなわちマイクロプログラム）で構成されており、該当するマイクロプログラムをマイクロコードＲＯＭ１４から読み出して実行することになる。なお、１つのプロセッサで利用できるインストラクション全体を「インストラクション・セット」と呼ぶ。

【００５６】マイクロコードＲＯＭ１４は、インストラクションの実行に必要なマイクロードを格納した読み出し専用メモリである。言い換えれば、マイクロコードＲＯＭ１４は、各インストラクションの実行手順を定義した不揮発メモリである。例えば、"ＬＯＡＤ"、"ＳＴＯＲＥ"、"ＭＯＶＥ"などのインストラクションは、ある変数をプロセッサ１００内のレジスタ（図示しない）に書き込むためのインストラクションであるが、これらの動作の詳細をマイクロコードＲＯＭ１４内で記述することによって定義することができる。

【００５７】メモリ管理ユニット（ＭＭＵ）１５は、プログラム・コードやデータをメイン・メモリ５０上に展開するなど、プロセッサ１００のメモリ空間を管理するためのユニットである。メモリ管理ユニット１５は、通常、仮想記憶機能とメモリ保護機能という２つの機能を提供している。仮想記憶機能とは、論理アドレスを物理アドレスに変換したり、物理アドレスに該当するブロック（ページ）がメイン・メモリ上に存在しないときに外部記憶装置との間でブロック（ページ）の入れ替え（すなわち「スワッピング」）を行う機能のことである。また、メモリ保護機能とは、１つのプログラムで生じた障害が他のプログラムに影響を及ぼさないようにする機能のことである。

【００５８】また、メモリ管理ユニット１５は、メモリ

空間上に存在する各プログラムのアドレス情報や属性情報を管理するためのＭＭＵテーブルを備えている。図２には、ＭＭＵテーブルの構造や機能を模式的に図解している。ＭＭＵテーブルの各エントリは、該当するプログラムのアドレスとフラグを格納する各フィールドを含んでいる。フラグはプログラムの属性情報（例えばプログラムが実行形式か、読み出し専用か、読み書き可能か）を示している。

【００５９】メモリ空間上の各プログラムは、夫々に固有のプログラムＩＤを有している。あるプログラムをプロセッサ１００に実行させるためには、メモリ管理ユニット１５にプログラムＩＤとそのアドレス情報・属性情報を与える旨のプログラム・コードを制御ユニット１３に供給すればよい。メモリ管理ユニット１５は、自身のＭＭＵテーブルを参照して、該当するプログラムの実行を許可する仕組みとなっている。下式は、あるプログラムの実行を許可するための疑似コードを記述したものである。

【００６０】

【数１】

```
Mov R1,Program_ID
（レジスタＲ１にプログラムＩＤをセットせよ）
Mov R2,Memory_Address_&Flag
（レジスタＲ２にアドレスとフラグをセットせよ）
Load MMU, R1,R2
（メモリ管理ユニットにレジスタＲ１とＲ２の内容をセットせよ）
Jmp Application
（アプリケーションにジャンプせよ）
```

【００６１】上記コード中の”Ｍｏｖ”や”Ｌｏａｄ”などは、プロセッサ１００に対するインストラクションであり、レジスタへの変数のセット等を意味する。各インストランションの詳細な実行手順は、ワイヤード・ロジック又はマイクロコードＲＯＭ１４で定義されたマイクロプログラムのいずれかの形態で定義されている（上述）。

【００６２】上記疑似コードのうち”Ｌｏａｄ　ＭＭＵ　…”なるステップは、実質上、アプリケーションの実行を許可するためのインストラクションである。本実施例では、この”Ｌｏａｄ　ＭＭＵ　…”インストラクションは、マイクロプログラムという形態で実装され、メモリ管理ユニット１５への変数セットの他に、所定の認証手続が含まれている。

【００６３】この認証手続は、マイクロプログラム中で記述されているので、プロセッサ１００の外部からは認証手続の存在自体が見えない。認証手続は、プロセッサ・チップ１００内部で完結的に実行されるので、この動作を外部から監視することは殆ど不可能である。したがって、”Ｌｏａｄ　ＭＭＵ　…”インストラクションを発行した外部プログラムからは全く見えない状態で、プロセッサ１００はこの認証手続を実行することができる。但し、認証手続の詳細は後述する。

【００６４】プロセッサＩＤ１６は、プロセッサ１００に固有の識別情報（通常はシリアルな番号で構成される）であり、例えば製造時に一義的に付与される。プロセッサＩＤ１６の実体は、不揮発メモリ（図示しない）に書き込まれた数値・文字データであり、制御ユニット１３によって適宜読み出される。

【００６５】暗号化復号化ユニット１７は、所定のデータを暗号化したり、逆に暗号化されたデータを復号化するための演算ユニットである。例えば、プロセッサＩＤ１６のような秘匿性の高いデータは、暗号化復号化ユニット１７によって予め暗号化してから、バス・インターフェース１１経由でプロセッサ１００の外部に出力される。逆に、暗号化されたデータ（例えばソフトウェア供給者から渡されるＫｅｙ：後述）の復号化も行う。

【００６６】暗号化及び復号化の方式は、共通鍵、公開鍵のうちいずれの方式を採用してもよい。後者の公開鍵方式は、公開鍵と秘密鍵という一組の暗号鍵が用いられ、一方の鍵で暗号化したときには他方の鍵でしか復号化できない。例えば、第３者が公開鍵で暗号化したデータは秘密鍵を持つ本人しか復号化できないので、秘密情報を安全に交換することができる（周知）。

【００６７】なお、プロセッサ１００を構成するためには、図１に示した以外にも多くのユニット等が必要である。図１から省略されたユニットの例は、整数同士の算術演算と論理演算（ＡＮＤ，ＯＲ，ＮＯＴ）を担当するＡＬＵ（Arithmetic and Logic Unit）や、実数（浮動小数点等）演算を担当するＦＰＵ（Floating-Point Unit）、インストラクションの実行手順を直線的に実装した配線論理回路（ＰＬＡ：Programmable Logic Array）、入出力用のレジスタなどである。但し、これらは当業者には周知であり、また、本発明の要旨を構成するものではないので、本明細書中では省略している。また、図面の錯綜を回避するため、図中の各ハードウェア・ブロック間の接続も一部しか図示していない点を了承されたい。

【００６８】次いで、プロセッサ１００によるプログラム保護動作について、図３を参照しながら説明する。

【００６９】ソフトウェア供給業者は、保護対象たるソフトウェア・プログラムをフロッピー・ディスク又はＣＤ－ＲＯＭなどの記憶媒体に格納した形態で、市場に流通する。このソフトウェア・プログラムには、固有の識

別情報すなわちソフトウェアＩＤ（通常はシリアルな番号で構成される）が付されている。ユーザは、フロッピー・ディスク又はＣＤ－ＲＯＭなどの記憶媒体をソフトウェア供給業者から購入するという形態で、ソフトウェア・プログラムの限定的な使用権を取得する（記憶媒体の購入は、ソフトウェァの著作権や無制限な使用権を全く意味しない点に充分留意されたい）。そして、ユーザは、記憶媒体を所定のドライブ・ユニット８１／８２に装填して、ソフトウェア・プログラムをシステム２００のハード・ディスク８１上にインストールする。

【００７０】このインストール作業には、プロセッサＩＤ１６を共通鍵により暗号化する手続（ステップＳ１００）と、暗号化したプロセッサＩＤをソフトウェア供給業者に送付する手続（ステップＳ１０２）を含んでいる。暗号化処理は、暗号化復号化ユニット１７を用いて行われる。送付手続は、ソフトウェア業者に対するＫｅｙ（後述）の要求を意味する。なお、送付手続は、郵送に依る他、ＮＩＣ９０で接続された外部ネットワーク経由での伝送という形態で行ってもよい。

【００７１】ソフトウェア供給業者は、受け取った暗号情報を復号化して、元のプロセッサＩＤを得る（ステップＳ１０４）。そして、このプロセッサＩＤと、業者自身が手元で管理しているソフトウェアＩＤとを一体化して、共通鍵により暗号化することによってＫｅｙを生成する（ステップＳ１０６）。Ｋｅｙは、プロセッサＩＤとソフトウェアＩＤとを１つのコードとして同時に暗号化したものであり、各々に切り離すことはできない。Ｋｅｙは、ソフトウェアの使用不能状態を解く鍵としての性質を持ち、郵送あるいはネットワーク経由でユーザに返送される（ステップＳ１０８）。

【００７２】このＫｅｙを取得した状態で、コンピュータ・システム２００上でソフトウェア・プログラムの実行が促されたとする。これは、例えばＯＳ（オペレーティング・システム）のような外部プログラムが”ＬＯＡＤ　ＭＭＵ…”のような実行許可インストラクションを発行して、メモリ管理ユニット１５にプログラムＩＤとプログラムの格納番地／属性情報がセットすることにより、実現される。

【００７３】本実施例の実行許可インストラクションに該当する処理手続は、図３の破線ブロックで囲まれている。実行許可は、メモリ管理ユニット１５への各変数のセット以外に所定の認証手続を含んでいる。

【００７４】当該インストラクションが発行されると、プロセッサ１００は、まず、受け取っているＫｅｙを復号化して、プロセッサＩＤとソフトウェアＩＤとを取得する（ステップＳ１１０）。復号化処理は、暗号化復号化ユニット１７を用いて行われる。

【００７５】次いで、プロセッサ１００は、復号化して得られたプロセッサＩＤとソフトウェアＩＤを、プロセッサ１００自身に内蔵されたプロセッサＩＤ、及び装填した記憶媒体から得られたソフトウェアＩＤの各々と比較する（ステップＳ１１２）。

【００７６】この比較の結果、照合に失敗したならば、メモリ管理ユニット１５によるプログラムの実行は拒否される（ステップＳ１１４）。他方、照合が成功裡に終わったならば、メモリ管理ユニット１５によるプログラムの実行は肯定され（ステップＳ１１６）、プログラムが実行される（ステップＳ１１８）。

【００７７】このような実行許可インストラクションの手続は、例えばマイクロコードＲＯＭ１４中で、上述のようなインストラクションの処理手続を記述することによって実現可能である。

【００７８】ステップＳ１１０～Ｓ１１６は、”Ｌｏａｄ　ＭＭＵ　…”という単一のインストラクションで実行される。該インストラクションを記述した疑似マイクロプログラム・コードを、以下に示しておく。

【００７９】

【数２】

```
mtspr IBAT0, Rs1, Rs2
（Ｒｓ１にプログラムのアドレスとフラグを、
Ｒｓ２にソフト供給業者から得たキーを、夫々セットする）
ld temp_reg1, Rs2
（キーを取得する）
decrypt temp_reg1
（プロセッサ１００に内蔵された共通鍵で復合化する）
cmp temp_reg1, temp_reg3
（ＩＤどうしを比較照合する）
jne unmatched
ld temp_reg1, Rs
（Ｒｓにプログラムのアドレスとフラグをセットする）
st IBAT0, temp_reg1
set flag_good
end_of_instruction
unmatched;
```

```
set flag_bad
end_of_instruction
```

【００８０】このようなマイクロプログラムに従った認証手続は、外部プログラムからは全く見えない状態で実行される。本実施例に係る認証手続は、プロセッサ・チップ１００内部で完結されるので、プロセッサ１００外部からの監視は殆ど不可能なのである。

【００８１】第２の実施の形態

本発明の第２の実施例は、図１に示したものと等価なハードウェア構成により具現される。但し、第１の実施例との相違点は、共通鍵ではなく公開鍵方式で暗号化する点にある。図４には、第２の実施例に係るプログラム保護動作をフローチャートにして示している。以下、図４を参照しながら説明する。

【００８２】ソフトウェア供給業者は、保護対象たるソフトウェア・プログラムをフロッピー・ディスク又はＣＤ－ＲＯＭなどの記憶媒体に格納した形態で、市場に流通する。このソフトウェア・プログラムには、固有の識別情報すなわちソフトウェアＩＤ（通常はシリアルな番号で構成される）が付されている。ユーザは、フロッピー・ディスク又はＣＤ－ＲＯＭなどの記憶媒体をソフトウェア供給業者から購入するという形態で、ソフトウェア・プログラムの限定的な使用権を取得する（記憶媒体の購入は、ソフトウェァの著作権や無制限な使用権を全く意味しない点に充分留意されたい）。そして、ユーザは、記憶媒体を所定のドライブ・ユニット８１／８２に装填して、ソフトウェア・プログラムをシステム２００のハード・ディスク８１上にインストールする。

【００８３】このインストール作業には、プロセッサＩＤ１６を暗号化する手続（ステップＳ２００）と、暗号化したプロセッサＩＤ及びプロセッサ１００に内蔵された公開鍵をソフトウェア供給業者に送付する手続（ステップＳ２０２）を含んでいる。暗号化処理は、暗号化復号化ユニット１７が、ソフトウェア供給業者から与えられた公開鍵を用いて行われる。送付手続は、ソフトウェア供給業者に対するＫｅｙ（後述）の要求を意味する。なお、送付手続は、郵送に依る他、ＮＩＣ９０で接続された外部ネットワーク経由での伝送という形態で行ってもよい。

【００８４】ソフトウェア供給業者は、受け取った暗号情報を、自分の秘密鍵を用いて復号化して、元のプロセッサＩＤを得る（ステップＳ２０４）。そして、このプロセッサＩＤと、業者自身が手元で管理しているソフトウェアＩＤとを一体化して、再びプロセッサ１００の公開鍵を用いて暗号化して、Ｋｅｙを生成する（ステップＳ２０６）。Ｋｅｙは、プロセッサＩＤとソフトウェアＩＤとを１つのコードとして同時に暗号化したものであり、各々に切り離すことはできない。Ｋｅｙは、ソフトウェアの使用不能状態を解く鍵としての性質を持ち、郵送あるいはネットワーク経由でユーザに返送される（ステップＳ２０８）。

【００８５】このＫｅｙを取得した状態で、コンピュータ・システム２００上でソフトウェア・プログラムの実行が促されたとする。これは、例えばＯＳ（オペレーティング・システム）のような外部プログラムが”ＬＯＡＤ　ＭＭＵ…”のような実行許可インストラクションを発行して、メモリ管理ユニット１５にプログラムＩＤとプログラムの格納番地／属性情報がセットすることにより、実現される。

【００８６】本実施例の実行許可インストラクションに該当する処理手続は、図４の破線ブロックで囲まれている。実行許可は、メモリ管理ユニット１５への各変数のセット以外に所定の認証手続を含んでいる。

【００８７】当該インストラクションが発行されると、プロセッサ１００は、受け取っているＫｅｙを復号化して、プロセッサＩＤとソフトウェアＩＤとを取得する（ステップＳ２１０）。復号化処理は、暗号化復号化ユニット１７が、プロセッサ１００自身に内蔵された秘密鍵を用いて行う。

【００８８】次いで、プロセッサ１００は、復号化して得られたプロセッサＩＤとソフトウェアＩＤを、プロセッサ１００自身に内蔵されたプロセッサＩＤ、及び装填した記憶媒体から得られたソフトウェアＩＤの各々と比較する（ステップＳ２１２）。

【００８９】この比較の結果、照合に失敗したならば、メモリ管理ユニット１５によるプログラムの実行は拒否される（ステップＳ２１４）。他方、照合が成功裡に終わったならば、メモリ管理ユニット１５によるプログラムの実行は肯定され（ステップＳ２１６）、プログラムが実行される（ステップＳ２１８）。

【００９０】このような実行許可インストラクションの手続は、例えばマイクロコードＲＯＭ１４中で、上述のようなインストラクションの処理手続を記述することによって実現可能である。

【００９１】ステップＳ２１０～Ｓ２１６は、”Ｌｏａｄ　ＭＭＵ　…”という単一のインストラクションで実行される。該インストラクションを記述した疑似マイクロプログラム・コードを、以下に示しておく。

【００９２】

【数３】

```
mtspr IBAT0, Rs1, Rs2
（Ｒｓ１にプログラムのアドレスとフラグを、
Ｒｓ２にソフト供給業者から得たキーを、夫々セットする）
ld temp_reg1
```

```
        （キーを取得する）
        decrypt temp_reg1
        （プロセッサ１００に内蔵された秘密鍵で復合化する）
        cmp temp_reg1, temp_reg3
        （ＩＤどうしを比較照合する）
        jne unmatched
        ld temp_reg1, Rs
        （Ｒｓにプログラムのアドレスとフラグをセットする）
        st IBAT0, temp_reg1
        set flag_good
        end_of_instruction
unmatched;
        set flag_bad
        end_of_instruction
```

【００９３】このようなマイクロプログラムに従った認証手続は、外部プログラムからは全く見えない状態で実行される。本実施例に係る認証手続は、プロセッサ・チップ１００内部で完結されるので、プロセッサ１００外部からの監視は殆ど不可能なのである。

【００９４】第３の実施の形態

本発明の第３の実施例は、プロセッサ１００が持つプロセッサＩＤを用いず、これに代わって時々刻々発生する乱数を用いるとともに、セキュリティ・デバイス９５との協働的動作によってプログラム保護を行う点で、上述の第１及び第２の実施例とは相違する。

【００９５】図５には、本発明の第３の実施の形態に係るコンピュータ・システム２００及びプロセッサ１００のハードウェア構成を模式的に示している。以下、各部について説明する。

【００９６】プロセッサ１００の外部ピンに直結したプロセッサ・バス１０１は、ブリッジ回路６０経由でシステム・バス７０に接続されている。ブリッジ回路６０は、プロセッサ・バス１０１とシステム・バス７０間の動作速度の相違を吸収するためのデータ・バッファや、メイン・メモリ５０へのアクセス動作を制御するためのメモリ・コントローラを含んでいる。

【００９７】メイン・メモリ５０は、プロセッサ１００が実行するプログラム・コードや処理データを一時格納するための読み書き可能メモリであり、通常は、１以上のＤＲＡＭ（ダイナミックＲＡＭ）チップで構成される。メイン・メモリ５０上で展開されるメモリ空間は、メモリ管理ユニット１５によって管理される。

【００９８】システム・バス７０には、ＰＣＩ（Peripheral Component Interconnect）バスやＩＳＡ（Industry Standard Architecture）バスが該当し、各種周辺装置が相互接続されている。周辺装置には、例えばキーボードやディスプレイなどのコンソール類（図示しない）、ハード・ディスク・ドライブ（ＨＤＤ）８１、フロッピー・ディスク・ドライブ（ＦＤＤ）８２、ＣＤ－ＲＯＭドライブ８３などの外部記憶装置類、外部ネットワークと接続するためのネットワーク・インターフェース・カード（ＮＩＣ）９０、シリアル・ポート経由でシリアル・データの入出力を行うためのシリアルＩ／Ｏコントローラ９１などが含まれる。なお、バス７０上の動作は、ロジック・アナライザなどを用いて解析することができる。

【００９９】本実施例では、シリアル・ポートにはセキュリティ・デバイス９５が外部接続されている。セキュリティ・デバイス９５は、保護対象たるソフトウェア・プログラムに付随する性質を持ち、ソフトウェアＩＤを格納する他、共通鍵又は公開鍵いずれかの方式による暗号化ユニットを含んでいる。セキュリティ・デバイス９５の動作については後述する。

【０１００】ソフトウェア供給業者は、通常、フロッピー・ディスクやＣＤ－ＲＯＭなどの記憶媒体の形態でソフトウェア・プログラムを配布する。各ユーザは、これら記憶媒体を所定のドライブ・ユニットに装填し、ＨＤＤ８１にコピーすることによって、ソフトウェア・プログラムをシステムにインストールすなわち利用可能な状態に置くようになっている。また、最近では、インターネットなどの外部ネットワーク経由でソフトウェア・プログラムが配布されることも多くなっている。

【０１０１】プロセッサ１００は、当業界においてＣＰＵ（Central Processing Unit）若しくはＭＰＵ（Micro Processing Unit）と呼ばれるものである。プロセッサ１００は、通常はマザー・ボード（図示しない）上に搭載される回路チップであり、各入出力装置（図示しない）を制御してデータの授受を行ったり、データを演算処理し、処理結果をメイン・メモリ５０上に記憶したりするなどのコンピュータ・システム（図示しない）の中枢機能を果たしている。

【０１０２】図５に示すように、本実施形態に係るプロセッサ１００は、バス・インターフェース１１と、デコーダ１２と、制御ユニット１３と、マイクロコードＲＯＭ１４と、メモリ管理ユニット１５と、暗号化復号化ユニット１７と、乱数発生ユニット１８とを含んでいる。

【０１０３】バス・インターフェース１１は、プロセッサ１００外部のコンポーネントとの間でインストラクションやデータを授受するためのユニットである。バス・インターフェース１１は、プロセッサ・バス１７と直結している。なお、プロセッサ・バス１７は、ブリッジ回路６０を経由してメイン・メモリ５０やシステム・バス７０と相互接続されている。

【０１０４】デコーダ１２は、バス・インターフェースで受け取ったインストラクションを解釈して、制御ユニット１３が理解できる形式に変換するためのユニットである。

【０１０５】制御ユニット１３は、インストラクションを実行する手順を整えるユニットである。実行するインストラクションは、ワイヤード・ロジック（配線論理）で実行するタイプと、マイクロ・プログラムで実行するタイプの２つに大別される。前者の場合は、インストラクションの実行制御に必要な論理を直線的に実装した順序回路が用いられる。また、後者の場合には、インストラクションはマイクロコード（又はピココード）と呼ばれるさらに細かいコードの集合（すなわちマイクロプログラム）で構成されており、該当するマイクロプログラムをマイクロコードＲＯＭ１４から読み出して実行することになる。なお、１つのプロセッサで利用できるインストラクション全体を「インストラクション・セット」と呼ぶ。

【０１０６】マイクロコードＲＯＭ１４は、インストラクションの実行に必要なマイクロードを格納した読み出し専用メモリである。言い換えれば、マイクロコードＲＯＭ１４は、各インストラクションの実行手順を定義した不揮発メモリである。例えば、”ＬＯＡＤ”、”ＳＴＯＲＥ”、”ＭＯＶＥ”などのインストラクションは、ある変数をプロセッサ１００内のレジスタ（図示しない）に書き込むためのインストラクションであるが、これらの動作の詳細をマイクロコードＲＯＭ１４内で記述することによって定義することができる。

【０１０７】メモリ管理ユニット（ＭＭＵ）１５は、プログラム・コードやデータをメイン・メモリ上に展開するなど、プロセッサ１００のメモリ空間を管理するためのユニットである。メモリ管理ユニット１５は、通常、仮想記憶機能とメモリ保護機能という２つの機能を提供している。仮想記憶機能とは、論理アドレスを物理アドレスに変換したり、物理アドレスに該当するブロック（ページ）がメイン・メモリ上に存在しないときに外部記憶装置との間でブロック（ページ）の入れ替え（すなわち「スワッピング」）を行う機能のことである。また、メモリ保護機能とは、１つのプログラムで生じた障害が他のプログラムに影響を及ぼさないようにする機能のことである。

【０１０８】また、メモリ管理ユニット１５は、メモリ空間上に存在する各プログラムのアドレス情報や属性情報を管理するためのＭＭＵテーブルを備えている。ＭＭＵテーブルの構造や機能は図２を参照した上記説明と略同一なので、ここでは説明を省略する。

【０１０９】あるプログラムをプロセッサ１００に実行させるためには、メモリ管理ユニット１５にプログラムＩＤとそのアドレス情報・属性情報を与える旨のプログラム・コードを制御ユニット１３に供給すればよい。一般には、”Ｌｏａｄ　ＭＭＵ…”なるインストラクションによって、実質上、アプリケーションの実行が許可される。

【０１１０】本実施例では、この”Ｌｏａｄ　ＭＭＵ…”インストラクションは、マイクロプログラムという形態で実装され、メモリ管理ユニット１５への各変数セットの他に、所定の認証手続が含まれている。この認証手続は、マイクロプログラム中で記述されているので、プロセッサ１００の外部からは認証手続の存在自体が見えない。認証手続は、プロセッサ・チップ１００内部で完結的に実行されるので、この動作を外部から監視することは殆ど不可能である。したがって、”Ｌｏａｄ　ＭＭＵ　…”インストラクションを発行した外部プログラムからは全く見えない状態で、プロセッサ１００はこの認証手続を実行することができる。但し、認証手続の詳細は後述する。

【０１１１】暗号化復号化ユニット１７は、所定のデータを暗号化したり、逆に暗号化されたデータを復号化するための演算ユニットである。例えば、プロセッサ１００が外部に出力する乱数のように秘匿性のあるデータは、暗号化復号化ユニット１７によって予め暗号化してから、バス・インターフェース１１経由でプロセッサ１００の外部に出力される。逆に、暗号化されたデータ（例えばソフトウェア供給者から渡されるＫｅｙ：後述）の復号化も行う。

【０１１２】暗号化及び復号化の方式は、共通鍵、公開鍵のうちいずれの方式を採用してもよい。後者の公開鍵方式は、公開鍵と秘密鍵という一組の暗号鍵が用いられ、一方の鍵で暗号化したときには他方の鍵でしか復号化できない。例えば、第３者が公開鍵で暗号化したデータは秘密鍵を持つ本人しか復号化できないので、秘密情報を安全に送付することができる（周知）。

【０１１３】乱数発生ユニット１８は、時々刻々乱数を発生するためのユニットであり、本実施例ではプロセッサＩＤの代わりに用いられる。乱数を用いるのは、固定値を用いたのでは、認証手続を複数回重ねて動作を監視することで見破られてしまう虞があるからである。

【０１１４】なお、プロセッサ１００を構成するためには、図５に示した以外にも多くのユニット等が必要である。図５から省略されたユニットの例は、整数同士の算術演算と論理演算（ＡＮＤ，ＯＲ，ＮＯＴ）を担当するＡＬＵ（Arithmetic and Logic Unit）や、実数（浮動小数点等）演算を担当するＦＰＵ（Floating-Point Uni

ｔ）、インストラクションの実行手順を直線的に実装した配線論理回路（ＰＬＡ：Programmable Logic Array）、入出力用のレジスタなどである。但し、これらは当業者には周知であり、また、本発明の要旨を構成するものではないので、本明細書中では省略している。また、図面の錯綜を回避するため、図中の各ハードウェア・ブロック間の接続も一部しか図示していない点を了承されたい。

【０１１５】次いで、プロセッサ１００によるプログラム保護動作について、図６を参照しながら説明する。

【０１１６】ソフトウェア供給業者は、保護対象たるソフトウェア・プログラムをフロッピー・ディスク又はＣＤ－ＲＯＭなどの記憶媒体に格納した形態で、市場に流通する。このソフトウェア・プログラムには、固有の識別情報すなわちソフトウェアＩＤ（通常はシリアルな番号で構成される）が付されている。ユーザは、フロッピー・ディスク又はＣＤ－ＲＯＭなどの記憶媒体をソフトウェア供給業者から購入するという形態で、ソフトウェア・プログラムの限定的な使用権を取得する（記憶媒体の購入は、ソフトウェァの著作権や無制限な使用権を全く意味しない点に充分留意されたい）。

【０１１７】本実施例が、上記第１及び第２の実施例と相違するのは、ソフトウェア・プログラムを格納した記憶媒体と一緒にセキュリティ・デバイス９５が配布される点である。ユーザは、ソフトウェア・プログラムをハード・ディスク８１上にインストールするともに、セキュリティ・デバイス９５をシリアル・ポートに外部接続する（ステップＳ３００）。この実施例では、Ｋｅｙの要求は、ソフトウェア供給業者ではなくセキュリティ・デバイス９５に対して行われる（後述）。

【０１１８】コンピュータ・システム２００上でソフトウェア・プログラムの実行が促されたとする。これは、例えばＯＳ（オペレーティング・システム）のような外部プログラムが”ＬＯＡＤ　ＭＭＵ…”のような実行許可インストラクションを発行して、メモリ管理ユニット１５にプログラムＩＤとプログラムの格納番地／属性情報がセットすることにより、実現される。

【０１１９】本実施例の実行許可インストラクションに該当する処理手続は、図６の破線ブロックで囲まれている。実行許可は、メモリ管理ユニット１５への各変数のセット以外に所定の認証手続を含んでいる。

【０１２０】当該インストラクションが発行されると、プロセッサ１００は、まず、乱数発生ユニット１８が発生した乱数をセキュリティ・デバイス９５に送ることで、Ｋｅｙの要求を行う（ステップＳ３０２）。乱数の送付はシステム・バス７０経由で行われる。但し、乱数は時々刻々変化する性質を持ち、同じ乱数は一回しか使用しないので、バス７０の動作をスヌープされてセキュリティが破られる可能性は極めて低い。

【０１２１】セキュリティ・デバイス９５は、受け取った乱数と、自身が格納するソフトウェアＩＤとを一体化して、共通鍵を用いて暗号化してＫｅｙを生成し、システム・バス７０経由でプロセッサ１００に送り返す（ステップＳ３０４）。

【０１２２】プロセッサ１００は、共通鍵を用いてＫｅｙを復号化して、乱数とソフトウェアＩＤとに分解する（ステップＳ３０６）。復号化処理は、暗号化復号化ユニット１７を用いて行われる。

【０１２３】次いで、プロセッサ１００は、復号化して得られた乱数とソフトウェアＩＤを、プロセッサ１００自身が保持する元の乱数、及び装填した記憶媒体から得られたソフトウェアＩＤの各々と比較する（ステップＳ３０８）。

【０１２４】この比較の結果、照合に失敗したならば、メモリ管理ユニット１５によるプログラムの実行は拒否される（ステップＳ３１０）。他方、照合が成功裡に終わったならば、メモリ管理ユニット１５によるプログラムの実行は肯定され（ステップＳ３１２）、プログラムが実行される（ステップＳ３１４）。

【０１２５】このような実行許可インストラクションの手続は、例えばマイクロコードＲＯＭ１４中で、上述のようなインストラクションの処理手続を記述することによって実現可能である。

【０１２６】ステップＳ３０２～Ｓ３１２は、”Ｌｏａｄ　ＭＭＵ　…”という単一のインストラクションで実行される。該インストラクションを記述した疑似マイクロプログラム・コードを、以下に示しておく。

【０１２７】

【数４】

```
mtspr IBAT0, Rs1, Rs2
（Ｒｓ１にプログラムのアドレスとフラグを、
Ｒｓ２にソフト供給業者から得たキーを、夫々セットする）
ld temp_reg1, Rs2
（ＩＤを取得する）
gen temp_reg2
（乱数を発生する）
end Cokey, temp_reg2
（共通鍵と乱数をセキュリティ・デバイスに送信する）
receive Encrypteddata, temp_reg2
```

```
        （暗号化されたＩＤをセキュリティ・デバイスから受信する）
        decrypt temp_reg3
        （ＩＤを共通鍵で復合化する）
        cmp temp_reg1, temp_reg3
        （ＩＤどうしを比較照合する）
        jne unmatched
        cmp temp_reg2, temp_reg3
        （乱数どうしを比較照合する）
        jne unmatch
        ld temp_reg1, Rs
        （Ｒｓにプログラムのアドレスとフラグをセットする）
        st IBAT0, temp_reg1
        set flag_good
        end_of_instruction
unmatched;
        set flag_bad
        end_of_instruction
```

【０１２８】このようなマイクロプログラムに従った認証手続は、外部プログラムからは全く見えない状態で実行される。本実施例に係る認証手続は、プロセッサ・チップ１００内部で完結されるので、プロセッサ１００外部からの監視は殆ど不可能なのである。

【０１２９】第４の実施の形態

本発明の第４の実施例は、図５に示したものと等価なハードウェア構成により具現される。但し、第３の実施例との相違点は、共通鍵ではなく公開鍵方式で暗号化する点にある。図７には、第４の実施例に係るプログラム保護動作をフローチャートにして示している。以下、図７を参照しながら説明する。

【０１３０】ソフトウェア供給業者は、保護対象たるソフトウェア・プログラムをフロッピー・ディスク又はＣＤ－ＲＯＭなどの記憶媒体に格納した形態で、市場に流通する。このソフトウェア・プログラムには、固有の識別情報すなわちソフトウェアＩＤ（通常はシリアルな番号で構成される）が付されている。ユーザは、フロッピー・ディスク又はＣＤ－ＲＯＭなどの記憶媒体をソフトウェア供給業者から購入するという形態で、ソフトウェア・プログラムの限定的な使用権を取得する（記憶媒体の購入は、ソフトウェアの著作権や無制限な使用権を全く意味しない点に充分留意されたい）。

【０１３１】本実施例では、ソフトウェア・プログラムを格納した記憶媒体と一緒にセキュリティ・デバイス９５に配布される。ユーザは、ソフトウェア・プログラムをハード・ディスク８１上にインストールするとともに、セキュリティ・デバイス９５をシリアル・ポートに外部接続する（ステップＳ４００）。Ｋｅｙの要求は、ソフトウェア供給業者ではなくセキュリティ・デバイス９５に対して行われる（後述）。

【０１３２】コンピュータ・システム２００上でソフトウェア・プログラムの実行が促されたとする。これは、例えばＯＳ（オペレーティング・システム）のような外部プログラムが”ＬＯＡＤ　ＭＭＵ…”のような実行許可インストラクションを発行して、メモリ管理ユニット１５にプログラムＩＤとプログラムの格納番地／属性情報がセットすることにより、実現される。

【０１３３】本実施例の実行許可インストラクションに該当する処理手続は、図７の破線ブロックで囲まれている。実行許可は、メモリ管理ユニット１５への各変数のセット以外に所定の認証手続を含んでいる。

【０１３４】当該インストラクションが発行されると、プロセッサ１００は、まず、乱数発生ユニット１８が発生した乱数とプロセッサ１００の公開鍵をセキュリティ・デバイス９５に送ることで、Ｋｅｙの要求を行う（ステップＳ４０２）。乱数の送付はシステム・バス７０経由で行われる。但し、乱数は時々刻々変化する性質を持ち、同じ乱数は一回しか使用しないので、バス７０の動作をスヌープされてセキュリティが破られる可能性は極めて低い。

【０１３５】セキュリティ・デバイス９５は、受け取った乱数と、自身が格納するソフトウェアＩＤとを一体化して、公開鍵を用いて暗号化してＫｅｙを生成し、システム・バス７０経由でプロセッサ１００に送り返す（ステップＳ４０４）。

【０１３６】プロセッサ１００は、自身の秘密鍵を用いてＫｅｙを復号化して、乱数とソフトウェアＩＤとに分解する（ステップＳ４０６）。復号化処理は、暗号化復号化ユニット１７を用いて行われる。

【０１３７】次いで、プロセッサ１００は、復号化して得られた乱数とソフトウェアＩＤを、プロセッサ１００自身が保持する元の乱数、及び装填した記憶媒体から得られたソフトウェアＩＤの各々と比較する（ステップＳ４０８）。

【０１３８】この比較の結果、照合に失敗したならば、

メモリ管理ユニット１５によるプログラムの実行は拒否される（ステップＳ４１０）。他方、照合が成功裡に終わったならば、メモリ管理ユニット１５によるプログラムの実行は肯定され（ステップＳ４１２）、プログラムが実行される（ステップＳ４１４）。

【０１３９】このような実行許可インストラクションの手続は、例えばマイクロコードＲＯＭ１４中で、このようにインストラクションの処理手続を記述することによって実現可能である。

【０１４０】ステップＳ４０２～Ｓ４１２は、”Ｌｏａｄ　ＭＭＵ　…”という単一のインストラクションで実行される。該インストラクションを記述した疑似マイクロプログラム・コードを、以下に示しておく。

【０１４１】

【数５】

```
                mtspr IBAT0, Rs1, Rs2
                （Ｒｓ１にプログラムのアドレスとフラグを、
                Ｒｓ２にソフト供給業者から得たキーを、夫々セットする）
                ld temp_reg1, Rs2
                （ＩＤを取得する）
                gen temp_reg2
                （乱数を発生する）
                end Pubkey, temp_reg2
                （公開鍵と乱数をセキュリティ・デバイスに送信する）
                receive Encrypteddata, temp_reg2
                （暗号化されたＩＤをセキュリティ・デバイスから受信する）
                decrypt temp_reg3
                （ＩＤを秘密鍵で復合化する）
                cmp temp_reg1, temp_reg3
                （ＩＤどうしを比較照合する）
                jne unmatched
                cmp temp_reg2, temp_reg3
                （乱数どうしを比較照合する）
                jne unmatched
                ld temp_reg1, Rs
                （Ｒｓにプログラムのアドレスとフラグをセットする）
                st IBAT0, temp_reg1
                set flag_good
                end_of_instruction
        unmatched;
                set flag_bad
                end_of_instruction
```

【０１４２】このようなマイクロプログラムに従った認証手続は、外部プログラムからは全く見えない状態で実行される。本実施例に係る認証手続は、プロセッサ・チップ１００内部で完結されるので、プロセッサ１００外部からの監視は殆ど不可能なのである。

【０１４３】追補

以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参酌すべきである。

【０１４４】

【発明の効果】以上詳記したように、本発明によれば、コンピュータ・システムに提供されるソフトウェア・プログラムの不正な使用・実行を防止する優れた技術を提供することができる。

【０１４５】また、本発明によれば、プログラムの不正実行防止機能を備えた優れたプロセッサ、プログラムの不正実行を防止するためにプロセッサが実行するインストラクション、及びプログラムの不正実行防止方法を提供することができる。

【０１４６】また、本発明によれば、外部から監視・変更することができないプログラムの不正実行防止機能付き優れたプロセッサ、プロセッサが実行するインストラクション、及びプログラムの不正実行防止方法を提供することにある。

【０１４７】また、本発明によれば、ソフトウェア供給者がソフトウェア・プログラムの使用権をただ１つのプ

ロセッサにのみ許可することができる、優れたプログラムの不正実行防止方法を提供することができる。

【図面の簡単な説明】

【図１】図１は、本発明の第１の実施の形態に係るコンピュータ・システム及びプロセッサ１００のハードウェア構成を模式的に示した図である。

【図２】図２は、ＭＭＵテーブルの構造や機能を模式的に示した図である。

【図３】図３は、本発明の第１の実施の形態に係るプログラム保護動作を示したフローチャートである。

【図４】図４は、本発明の第２の実施の形態に係るプログラム保護動作を示したフローチャートである。

【図５】図５は、本発明の第３の実施の形態に係るコンピュータ・システム及びプロセッサ１００のハードウェア構成を模式的に示した図である。

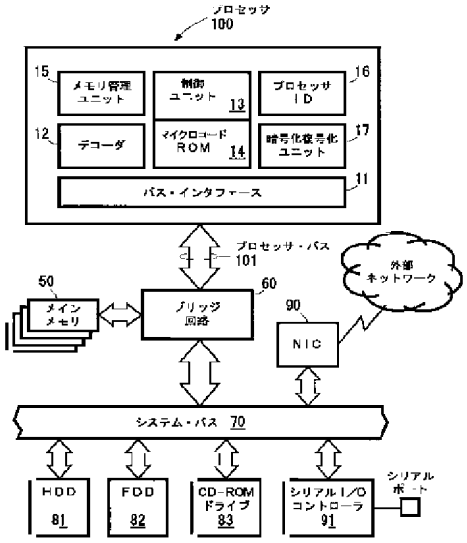
【図６】図６は、本発明の第４の実施の形態に係るプログラム保護動作を示したフローチャートである。

【図７】図７は、本発明の第５の実施の形態に係るプログラム保護動作を示したフローチャートである。

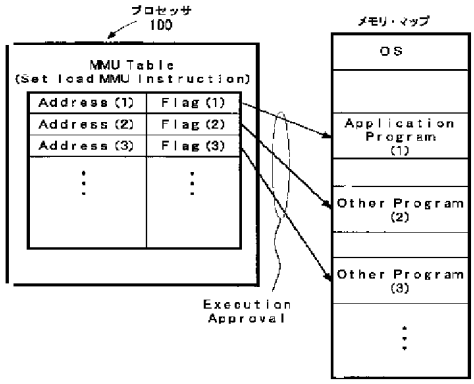
【符号の説明】

１１…バス・インターフェース、１２…デコーダ、１３…制御ユニット、１４…マイクロコードＲＯＭ、１５…メモリ管理ユニット（ＭＭＵ）、１６…プロセッサＩＤ、１７…暗号化復号化ユニット、５０…メイン・メモリ、６０…ブリッジ回路、７０…システム・バス、８１…ハード・ディスク・ドライブ（ＨＤＤ）、８２…フロッピー・ディスク・ドライブ（ＦＤＤ）、８３…ＣＤ－ＲＯＭドライブ、９０…ネットワーク・インターフェース・カード（ＮＩＣ）、９１…シリアルＩ／Ｏコントローラ、９５…セキュリティ・デバイス、１００…プロセッサ、２００…コンピュータ・システム。

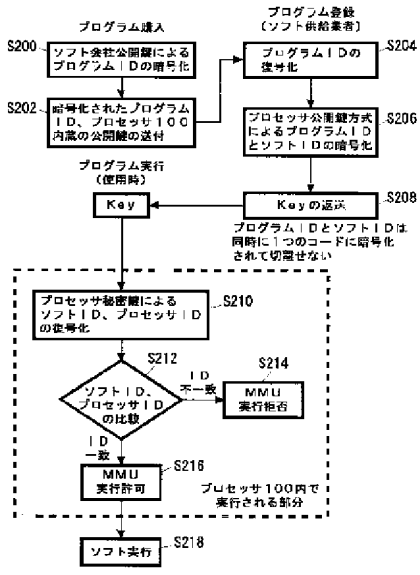
【図１】



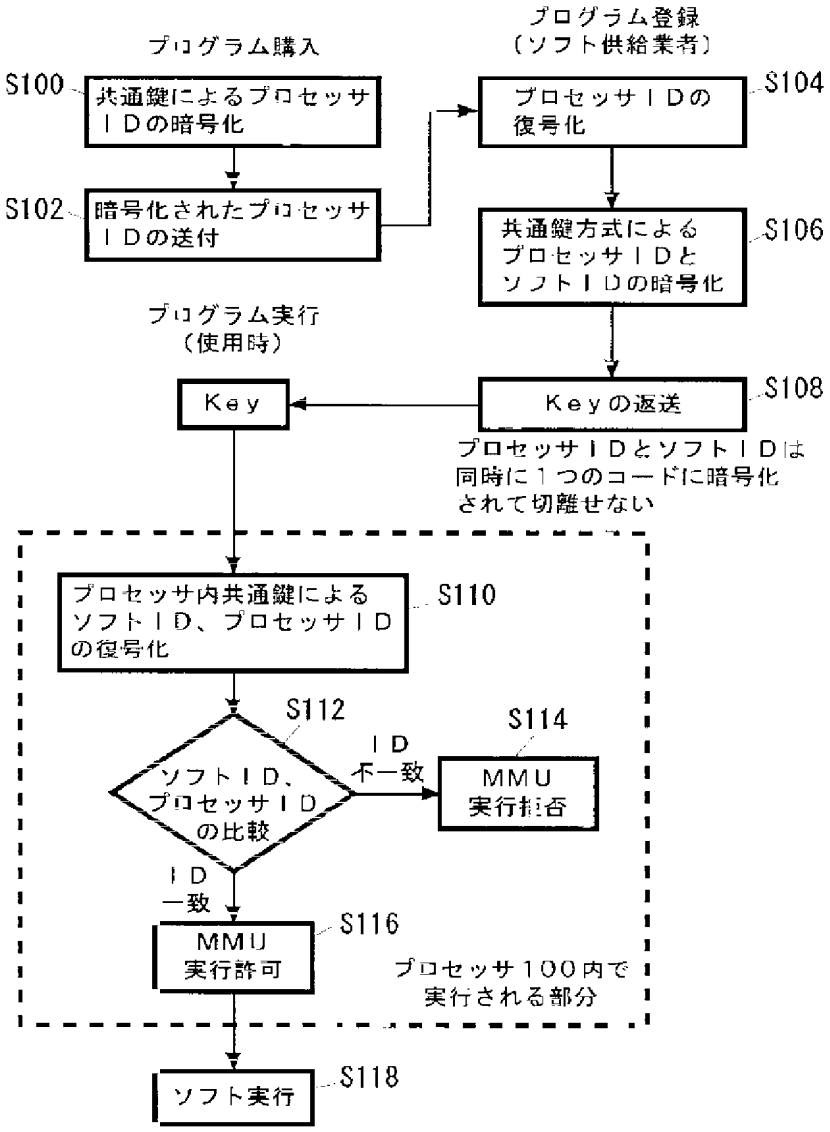
【図２】



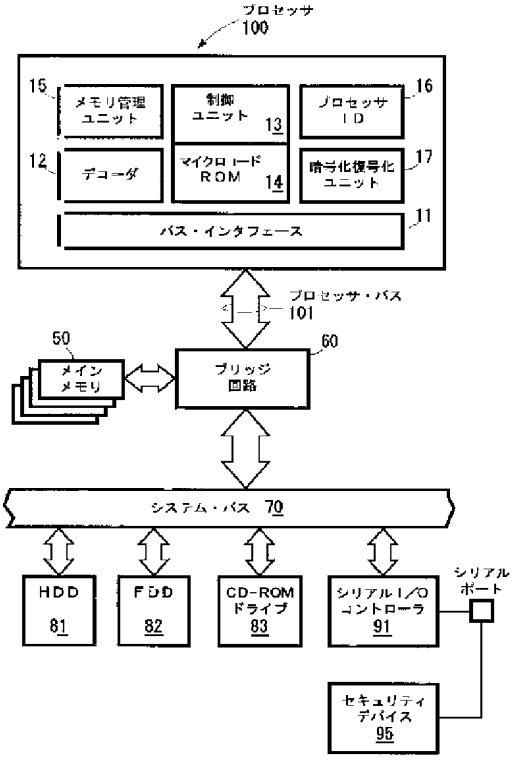
【図４】



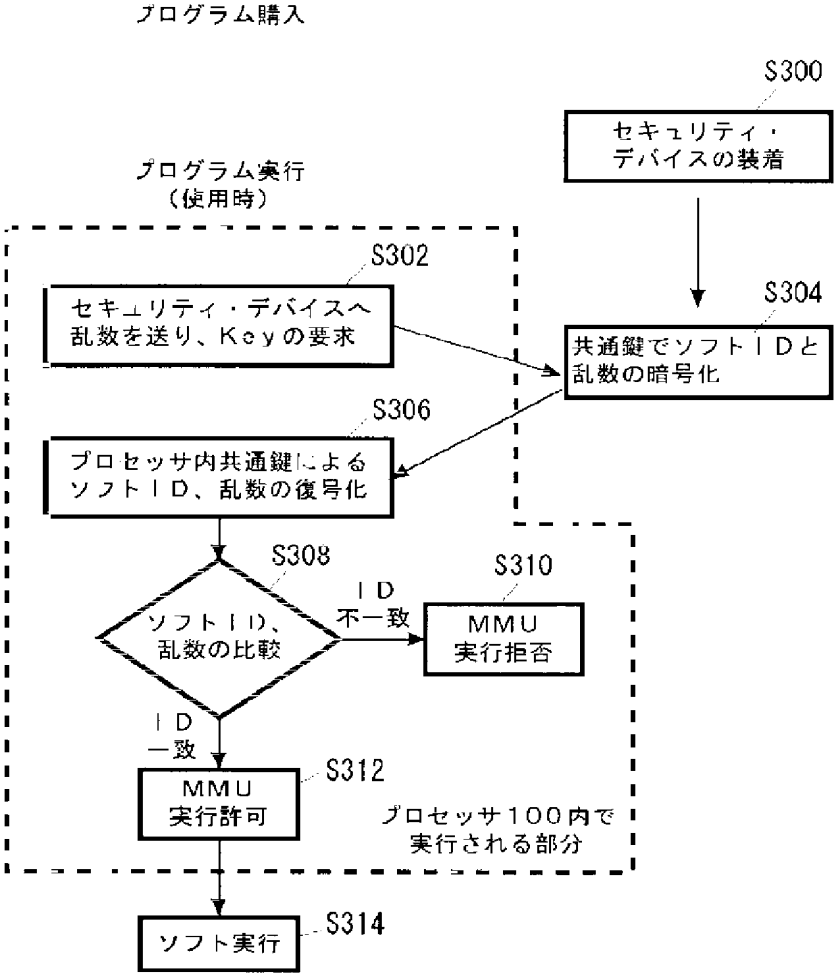
【図３】


プログラム購入
プログラム登録
（ソフト供給業者）
S100
共通鍵によるプロセッサＩＤの暗号化
S102
暗号化されたプロセッサＩＤの送付
S104
プロセッサＩＤの復号化
S106
共通鍵方式によるプロセッサＩＤとソフトＩＤの暗号化
プログラム実行
（使用時）
Key
S108
Keyの返送
プロセッサＩＤとソフトＩＤは同時に１つのコードに暗号化されて切離せない
プロセッサ内共通鍵によるソフトＩＤ、プロセッサＩＤの復号化
S110
S112
ソフトＩＤ、プロセッサＩＤの比較
ＩＤ不一致
S114
MMU実行拒否
ＩＤ一致
S116
MMU実行許可
プロセッサ１００内で実行される部分
S118
ソフト実行

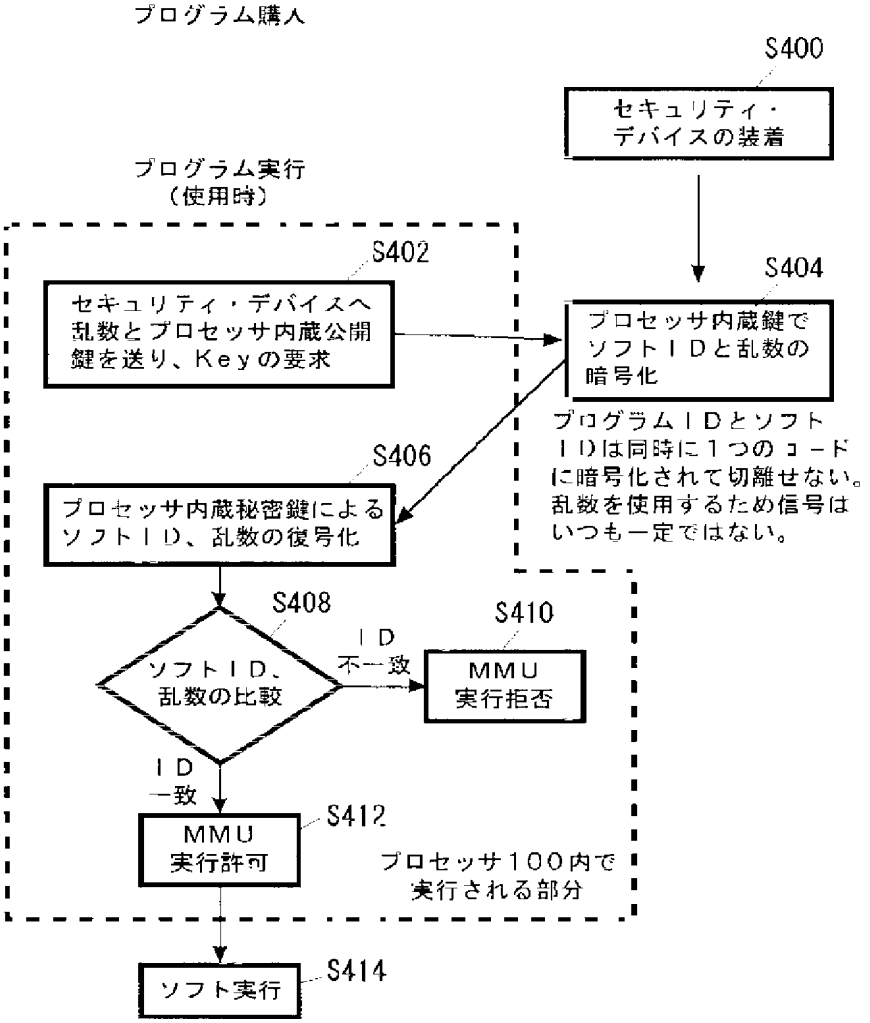
【図５】


プロセッサ
100
15
メモリ管理ユニット
制御ユニット
13
プロセッサID
16
12
デコーダ
マイクロコードROM
14
暗号化復号化ユニット
17
バス・インタフェース
11
プロセッサ・バス
101
50
メインメモリ
ブリッジ回路
60
システム・バス 70
HDD
81
FDD
82
CD-ROMドライブ
83
シリアルI/Oコントローラ
91
シリアルポート
セキュリティデバイス
95

【図６】


プログラム購入
S300
セキュリティ・
デバイスの装着
プログラム実行
（使用時）
S302
セキュリティ・デバイスへ
乱数を送り、Ｋｅｙの要求
S304
共通鍵でソフトＩＤと
乱数の暗号化
S306
プロセッサ内共通鍵による
ソフトＩＤ、乱数の復号化
S308
ソフトＩＤ、
乱数の比較
ＩＤ
不一致
S310
ＭＭＵ
実行拒否
ＩＤ
一致
S312
ＭＭＵ
実行許可
プロセッサ１００内で
実行される部分
S314
ソフト実行

【図７】


プログラム購入
S400
セキュリティ・
デバイスの装着
プログラム実行
（使用時）
S402
セキュリティ・デバイスへ
乱数とプロセッサ内蔵公開
鍵を送り、Ｋｅｙの要求
S404
プロセッサ内蔵鍵で
ソフトＩＤと乱数の
暗号化
プログラムＩＤとソフト
ＩＤは同時に１つのコード
に暗号化されて切離せない。
乱数を使用するため信号は
いつも一定ではない。
S406
プロセッサ内蔵秘密鍵による
ソフトＩＤ、乱数の復号化
S408
ソフトＩＤ、
乱数の比較
ＩＤ
不一致
S410
MMU
実行拒否
ＩＤ
一致
MMU
実行許可
S412
プロセッサ１００内で
実行される部分
ソフト実行
S414